

Multi-Objective Differential Evolution (MODE): An Evolutionary Algorithm for Multi-Objective Optimization Problems (MOOPs)

B. V. Babu* and B. Anbarasu
Birla Institute of Technology and Science (BITS)
PILANI – 333 031 (Rajasthan) INDIA
bvbabu@bits-pilani.ac.in

Abstract

The last decade has seen a surge of research activity on Multi-objective Optimization using evolutionary computation and a number of algorithms have been developed. Although Evolutionary Algorithms (EAs) are successful, to some extent, in solving Multi-Objective Optimization Problems (MOOPs), the methods appearing in the literature vary a lot in terms of their solutions and the way of comparing their best results with other existing algorithms. In other words, there is no well-accepted method for MOOPs that will produce a good set of solutions for all problems. This motivates the further development of good approaches to MOOPs. In this paper, we propose a novel approach using Differential Evolution (DE) for MOOPs, referred to herein as Multi-Objective Differential Evolution (MODE). The performance of MODE has been compared to that of other Elitist multi-objective evolutionary algorithms such as NSGA-II and its adapted form, NSGA-II-JG on four test problems. It has been observed that even though the computational complexity of MODE is high, the solutions obtained on the pareto optimal front are well diversified.

Index Terms — Evolutionary Computation, Differential Evolution, Evolutionary Multi-objective Optimization (EMO), Multi-Objective Differential Evolution (MODE).

1. Introduction

Till the last decade, most of the problems solved in the field of optimization involved only a single objective function. In the past several years, there has been an increasing interest in applying evolutionary algorithms to multiobjective optimization problems (MOOPs), since real world optimization problems often involve several conflicting objectives.

In such multiobjective studies, we often obtain a Pareto set of non-dominating (equally good) solutions, and a decisionmaker needs to use his intuition or additional information to decide upon the preferred solution. Indeed, Deb [1] has given several examples in different fields that are better studied using multiple objectives.

2. Elitist Multi-Objective Evolutionary Algorithms

During the past few years, a number of different Evolutionary Algorithms (EAs) were suggested to solve multi-objective optimization problems. An elitist non-dominated sorting genetic algorithm (NSGA-II) was proposed by Deb et al. [2]. NSGA-II has several advantages over the currently available multi-objective optimization algorithms. These have been reviewed by Deb [1]. NSGA-II uses the concept of elitism, borrowed from nature. Two main features of the algorithm are as follows: i) assigning fitness to population members based on nondominated sorting and ii) preserving diversity among solutions of the same nondominated front. In this algorithm, the better parents are given a chance to be part of the next generation. In contrast, the likelihood of this happening in the earlier algorithm, NSGA-I, that did not incorporate this concept, is very small. Unfortunately, the diversity decreases because of this elitism, and this needs to be counteracted by some means. One such adaptation, inspired by the concept of jumping genes (JG, or transposons) in biology, was developed by Kasat and Gupta [3]. This is referred to as NSGA-II-JG. This adaptation exploits the benefits of elitism, while still maintaining genetic diversity.

In this study, Differential Evolution (a highly efficient optimization algorithm for single objective problems), has been adapted for solving optimization problems involving multiple objectives. It has been applied to four different test problems chosen from Deb [1] and the results have been compared to that obtained using NSGA-II and NSGA-II-JG. The results encourage

*Corresponding Author:

Assistant Dean-ESD & Head-Chemical Engineering Department
BITS, PILANI-333 031 (Rajasthan) India
Phone: 0091-1596-245073 Ext. 205/224; Fax: 0091-1596-244183;
E-mail: bvbabu@bits-pilani.ac.in
Homepage: <http://discovery.bits-pilani.ac.in/discipline/chemical/BVb/>

the application of MODE to more complex and real-world multi-objective optimization problems.

3. Differential Evolution

Differential Evolution (DE) is a type of evolutionary algorithm originally proposed by Price and Storn [4] for optimization problems over a continuous domain. DE is similar to a (μ, λ) evolution strategy in which mutation plays the key role. The basic idea of DE is to adapt the search during the evolutionary process. At the start of evolution, the perturbations are large since parent individuals are far away from each other. As the evolutionary process matures, the population converges to a small region and the perturbations adaptively become small. As a result, the evolutionary algorithm performs a global exploratory search during the early stages of the evolutionary process and local exploitation during the mature stage of the search.

In DE, a solution, l , in a generation is a multi-dimensional vector $\bar{x}_{G=i}^l = (x_1, \dots, x_N)^T$. A population, $P_{G=k}$, at generation $G=k$ is a vector of M solutions ($M > 4$). The initial population, $P_{G=0} = \{x_{G=0}^1, \dots, x_{G=0}^M\}$, is initialized as

$$x_{i,G=0}^l = lower(x_i) + rand_i[0,1] \times (upper(x_i) - lower(x_i)),$$

$$l = 1, \dots, M \quad i = 1, 2, \dots, N$$

where, M is the population size, N is the solution's dimension, and each variable i in a solution vector l in the initial generation $G = 0$, $\bar{x}_{i,G=0}^l$, is initialized within its boundaries ($lower(x_i), upper(x_i)$). Selection is carried out to select four different solutions indices $r_1; r_2; r_3;$ and $j \in [1, M]$. The values of each variable in the child are changed with some crossover probability, CR , to

$$\forall i \leq N, x'_{i,G=k} = \begin{cases} x_{i,G=k-1}^{r_3} + F \times (x_{i,G=k-1}^{r_1} - x_{i,G=k-1}^{r_2}) & \text{if } (random[0,1] < CR \wedge i = i_{rand}) \\ x_{i,G=k-1}^j & \text{otherwise} \end{cases}$$

where, $F \in (0, 1)$ is a problem parameter representing the amount of perturbation added to the main parent. The new solution replaces the old one if it is better than the original and at least one of the variables should be changed. The latter is represented in the algorithm by randomly selecting a variable, i_{rand} ($1, N$). After crossover, if one or more of the variables in the new solution are outside their boundaries, the following repair rule is applied.

$$x'_{i,G=k} = \begin{cases} \frac{x_{i,G}^j + lower(x_i)}{2} & \text{if } x_{i,G+1}^j < lower(x_i) \\ lower(x_i) + \frac{x_{i,G}^j - upper(x_i)}{2} & \text{if } x_{i,G+1}^j > upper(x_i) \\ x_{i,G+1}^j & \text{otherwise} \end{cases}$$

The detailed algorithm is available in literature [8,9].

4. Different Strategies of DE

Price & Storn [4] gave the working principle of DE with single strategy. Later on, they suggested ten different strategies of DE. The strategies can vary based on the vector to be perturbed, number of difference vectors considered for perturbation, and finally the type of crossover used [4,8]. However, strategy-7 (DE/rand/1/bin) is the most successful and the most widely used strategy. The key parameters of control in DE are: NP-the population size, CR-the crossover constant, and F-the weight applied to random differential (scaling factor). In addition, some new strategies have been proposed and successfully applied to optimization of extraction process by Babu & Angira [6]. DE has been successfully applied in various fields [7-22].

5. Multi-Objective Differential Evolution

This is inspired from elitist nondominated sorting genetic algorithm (NSGA-II). In a single-objective optimization problem, the best solution, integral to the DE search process, may be easily identified by selecting the individual with highest fitness value. However, in a multi-objective domain, the goal is to identify the Pareto optimal solution set. In this proposed multi-objective differential evolution (MODE), a Pareto-based approach is introduced to implement the selection of the best individuals. Firstly, a population of size, NP, is generated randomly and the fitness functions are evaluated. At a given generation of the evolutionary search, the population is sorted into several ranks based on dominance concept. Secondly, DE operations are carried out over the individuals of the population. The fitness functions of the trial vectors, thus formed, are evaluated. One of the major differences between DE and MODE is that the trial vectors are not compared with the corresponding parent vectors. Instead, both the parent vectors and the trial vectors are combined to form a global population of size, $2 * NP$. Then, the ranking of the global population is carried out followed by the crowding distance calculation. The best NP individuals are selected based on its ranking and crowding distance. These act as the parent vectors for the next generation.

The procedure is carried out until the entire selected best NP individuals have a rank of one. The Pseudocode for MODE algorithm is presented below:

The following assumes that we are minimizing all the objective functions, f_q

- (1) Generate box, P , of N_p parent vectors using a random-number code to generate the several real variables. These vectors are given a sequence (position) number as generated
- (2) Classify these vectors into fronts based on non-domination [1], as follows:
 - a. Create new (empty) box, P' , of size, N_p
 - b. Transfer i^{th} vector from P to P' , starting with $i=1$
 - c. Compare vector I with each member, say each member, say j , already present in P' , one at a time
 - d. If i dominates³¹ over j (i.e. i is superior to or better than j in terms of all objective functions), remove the j^{th} vector from P' and put it back in its original location in P
 - e. If i dominated over by j , remove i from P' and put it back in its position in P
 - f. If i and j are non-dominating (i.e. there is at least one objective function associated with i that is superior to/better than that of j), keep both i and j in P' (in sequence). Test for all j present in P'
 - g. Repeat for next vector (in the sequence, without going back) in P till all N_p are tested. P' now contains a sub-box (of size $\leq N_p$) of non-dominated vectors (a subset of P), referred to as the first front or sub-box. Assign it a rank number, I_{rank} of 1
 - h. Create subsequent fronts in (lower) sub-boxes of P' , using Step 2b above (with the vectors remaining in P). Compare these members only with the members present in the current sub-box, and not with those in earlier (better) sub-boxes. Assign these $I_{\text{rank}}=2,3,\dots$ Finally, we have all N_p vectors in P' , boxed into one or more fronts
- (3) Spreading out: Evaluate the crowding distance, $I_{i,\text{dist}}$, for the i^{th} vector in any front, j , of P' using the following procedure:
 - a. Rearrange all vectors in front j in ascending order of the values of any one (say, the q^{th}) of their several objective functions (fitness functions). This provides a sequence, and, thus, defines the nearest neighbors of any vector in front j .
 - b. Find the largest cuboid (rectangle for two fitness functions) enclosing vector i that just touches its nearest neighbors in the f -space.
 - c. $I_{i,\text{dist}} \equiv \frac{1}{2} * (\text{sum of all sides of this cuboid})$

- d. Assign large values of $I_{i,\text{dist}}$ to solutions at the boundaries (the convergence characteristics would be influenced by this choice)
- (4) Perform DE operation over the NP target vectors in P' to generate NP trial vectors and store it in P'' .
 - a. Create new (empty) box, P'' , of size, N_p
 - b. Select a target vector, i in P' , starting with $i=1$
 - c. Choose two vectors, $r1$ and $r2$ at random from the NP vectors in P' and find the weighted difference. This is carried out by the following steps: 1) Generate two random numbers 2) decide which two population members are to be selected 3) Find the vector difference between the two vectors. Multiply this difference with F to obtain the weighted difference.
 - d. Find the noisy random vector. This is done by 1) Generate a random number 2) choose a third random vector, $r3$, from the NP vectors in P' 3) Add this vector to the weighted difference to obtain the noisy random vector
 - e. Perform Crossover between the target vector and noisy random vector to find the trial vector and put it in box P'' . This is carried out by 1) Generate random numbers equal to the dimension of the problem 2) For each of the dimensions: if random no. $> CR$; copy the value from the target vector, else copy the value from the noisy random vector into the trial vector and put it in box P''
- (5) Elitism: Copy all the N_p parent vectors (P') and all the N_p trial vectors (P'') into box PT . Box PT has $2N_p$ vectors
 - a. Reclassify these $2N_p$ vectors into fronts (box PT') using only non-domination (as described in Step 2 above)
 - b. Take the best N_p from box PT' and put into box P''' . The following procedure is adopted to identify the better of the two chromosomes. Chromosome i is better than chromosome j if

$$I_{i,\text{rank}} \neq I_{j,\text{rank}} : I_{i,\text{rank}} < I_{j,\text{rank}}$$

$$I_{i,\text{rank}} = I_{j,\text{rank}} : I_{i,\text{dist}} > I_{j,\text{dist}}$$

This completes one generation. Stop if appropriate criteria are met, e.g., the generation number $>$ maximum number of generations (user specified). Else, Copy P''' into starting box, P . Go to Step 2 above

6. Test Problems

The test problems were taken from Deb [1]. We first describe the test problems used to compare different algorithms.

Problem 1:

$$\text{Min. } f_1(x) = x_1; \quad 0.1 \leq x_1 \leq 1$$

$$\text{Min. } f_2(x) = (1+x_2)/x_1; \quad 0 \leq x_2 \leq 5$$

Problem 2:

$$\begin{aligned} \text{Max. } f_1(x) &= 1-x_1; & 0.1 \leq x_1 \leq 1 \\ \text{Max. } f_2 &= 60-(1+x_2)/x_1; & 0 \leq x_2 \leq 5 \end{aligned}$$

Problem 3:

$$\text{Min } f_1(d, l) = \rho \frac{\pi d^2}{4} l \quad 10 \leq d \leq 50 \text{mm}$$

$$\text{Min } f_2(d, l) = \frac{64PI^3}{3E\pi d^4} \quad 200 \leq l \leq 1000 \text{mm}$$

subject to

$$\sigma_{\max} \leq S,$$

$$\delta \leq \delta_{\max}$$

where, the maximum stress is calculated as follows :

$$\sigma_{\max} = \frac{32PI}{\pi d^3}$$

The following parameter values are used:

$$\rho=7800 \text{ kg/cu. m} \quad P = 1000 \text{ N} \quad E = 207 \text{ GPa}$$

$$S_y=300 \text{ MPa} \quad \delta_{\max} = 5 \text{ mm}$$

Problem 4:

$$\text{Min } f_1(x) = 0.5(x_1^2 + x_2^2) + \sin(x_1^2 + x_2^2)$$

$$\text{Min } f_2(x) = \frac{(3x_1 - 2x_2 + 4)^2}{8} + \frac{(x_1 - x_2 + 1)^2}{27} + 15$$

$$\text{Min } f_3(x) = \frac{1}{(x_1^2 + x_2^2 + 1)^2} - 1.1e^{-(x_1^2 - x_2^2)}$$

$$-3 \leq x_1, x_2 \leq 3$$

7. Results and Discussion

Several cases have been studied using the test problems. This section deals with the results obtained for various cases and the related inferences.

7.1 Case Studies

Table-1 shows the details of the Case-I to Case-VIII.

Single point Cross-over is used.

Mutation Probability for NSGA-II & NSGA-II-JG - 0.05

Table-1 Details of Cases I to VIII

Case	Problem No.	Type/ DE strategy	Accuracy	CR	N _p	Random Seed
Case I	1	Unconstrained DE/rand/1/bin	$x_1-0.001$ $x_2-0.001$	0.95	-	0.7534
Case II	2	Unconstrained DE/rand/1/bin	$x_1-0.001$ $x_2-0.001$	0.95	-	0.6985414
Case III	3	Constrained DE/rand/1/bin	$d-0.0001$ $l-0.001$	0.95	-	0.28446
Case IV	1	Unconstrained DE/rand/1/bin	$d-0.0001$ $l-0.001$	-	100	0.7534
Case V	3	Constrained DE/rand/1/bin	$d-0.0001$ $l-0.001$	-	100	0.28446
Case VI	3	Constrained DE/rand/1/exp	$d-0.0001$ $l-0.001$	-	100	0.28446
Case VII	3	Constrained DE/current-to-rand/1/bin	$d-0.0001$ $l-0.001$	-	100	0.28446
Case VIII	4	Unconstrained DE/rand/1/bin	$x_1-0.001$ $x_2-0.001$	-	100	0.35652

7.2 Results

The results obtained for various cases are presented in the form of tables (Tables 2-5).

Table 2 Effects of NP and other parameters on different Algorithms for Case-I

Algorithm	Parameter	Pop = 20		Pop = 40		Pop = 100	
		Gen	Time, ms	Gen	Time, ms	Gen	Time, ms
NSGA-II	-	10	672	7	500	7	563
NSGA-II-JG (Pjump)	0.5	7	468	10	718	7	578
	0.6	7	469	8	578	8	672
	0.7	8	546	7	500	9	734
	0.8	9	625	9	641	9	734
MODE (Scaling Factor-F)	0.9	8	547	8	562	9	750
	0.5	8	516	11	781	11	875
	0.6	6	406	9	641	12	953
	0.7	6	406	9	641	11	875
	0.8	11	750	10	703	14	1110
	0.9	6	406	11	766	15	1187

Table 3 Effects of CR and F on Generations in MODE for Case IV & Case V

Parameter	Generation for Case IV					Generation for Case V					
	Scaling Factor, F					Scaling Factor, F					
	0.5	0.6	0.7	0.8	0.9	0.5	0.6	0.7	0.8	0.9	
CR	0.5	11	11	10	11	11	11	12	12	11	11
	0.6	10	11	10	11	11	11	10	12	13	13
	0.7	11	13	11	11	12	12	13	13	12	14
	0.8	11	11	12	12	13	12	13	14	13	13
	0.9	11	12	14	13	13	13	12	13	14	15

Table 4 Effects of NP and other parameters on different algorithms for Case II

Algorithm	Parameter	Pop = 20		Pop = 40		Pop = 100	
		Gen	Time, ms	Gen	Time, ms	Gen	Time, ms
NSGA-II	-	5	328	8	563	8	688
NSGA-II-JG (Pjump)	0.5	5	344	8	563	8	656
	0.6	6	422	6	422	10	828
	0.7	5	344	7	500	10	813
	0.8	6	406	8	578	9	734
MODE (Scaling Factor-F)	0.9	6	407	7	500	9	734
	0.5	9	625	11	734	11	859
	0.6	8	547	10	703	11	859
	0.7	7	469	11	766	11	860
	0.8	9	609	11	766	14	1109
	0.9	6	406	10	703	13	1031

Table 5 Effects of NP and Other parameters on different Algorithms for CASE III

Algorithm	Parameter	Pop = 20		Pop = 40		Pop = 100	
		Gen	Time, ms	Gen	Time, ms	Gen	Time, ms
NSGA-II	-	7	485	6	437	8	687
NSGA-II-JG (Pjump)	0.5	5	328	8	578	8	687
	0.6	9	625	8	578	10	844
	0.7	7	485	8	563	9	767
	0.8	7	485	7	516	10	859
	0.9	6	422	10	719	10	859
MODE (Scaling Factor-F)	0.5	7	469	11	782	13	1078
	0.6	8	546	10	703	14	1156
	0.7	7	469	11	782	13	1078
	0.8	7	468	13	938	15	1234
	0.9	9	609	10	703	16	1312

7.3 Discussion

The inferences obtained from the results are discussed in terms of the following: population size, number of generation, controlling parameters and different strategies.

Effect of Population Size:

It can be observed from Figs. 1-3 that NSGA-II could converge to the global optimum even with a population size of 20 as compared to NSGA-II-JG and MODE.

However, the solutions obtained using NSGA-II are less diverse. For all the three algorithms, computational complexity increases with the population size. For a population size of 100, all the three algorithms converged to the Pareto set. Hence in most of the Multi-objective Optimization Problems (MOPs), the population size is taken as 100. The number of generations required for convergence is presented in Table 2, Table-4 and Table-5.

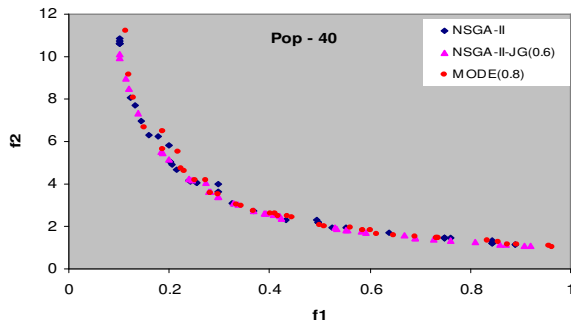


Fig. 1(a). Effects of NP and other parameters on different algorithms for Case I

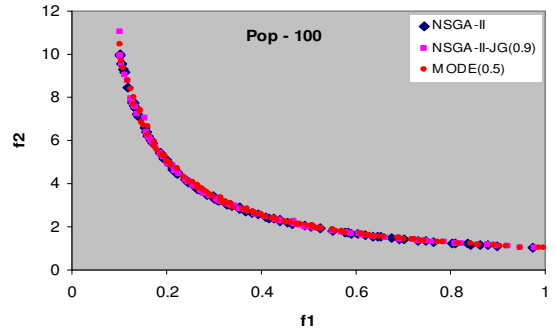


Fig. 1(b). Effects of NP and other parameters on different algorithms for Case I

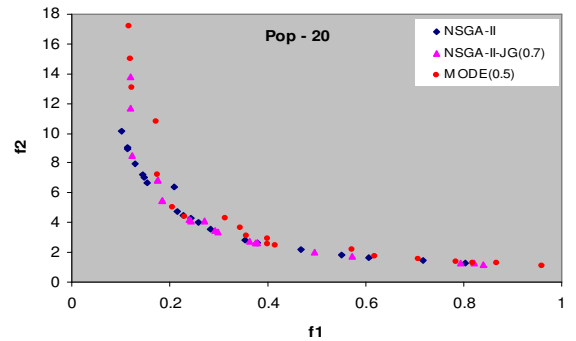


Fig. 1(c). Effects of NP and other parameters on different algorithms for Case I

Effect of Number of Generations:

Once the entire population reaches the first front, a slight increase in the number of generation increases the diversity of the solution. However, for a very high value of generation the diversity decreases and the computational time increases.

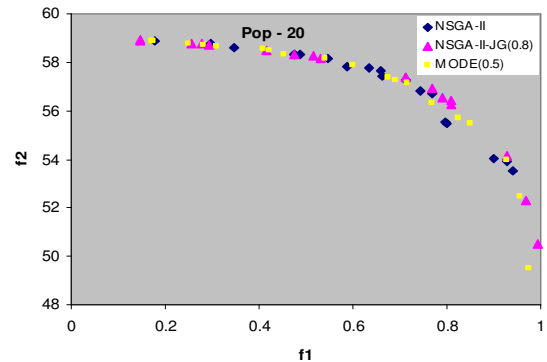


Fig. 2(a). Effects of NP and other parameters on different algorithms for CaseII

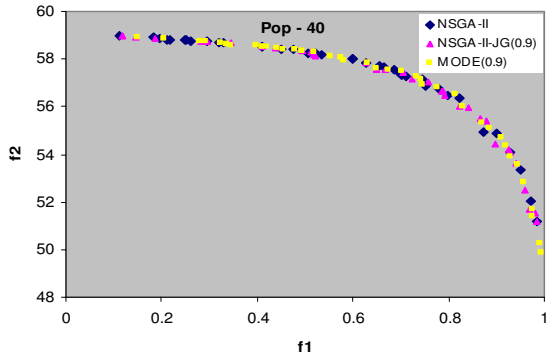


Fig. 2(b). Effects of NP and other parameters on different algorithms for CaseII

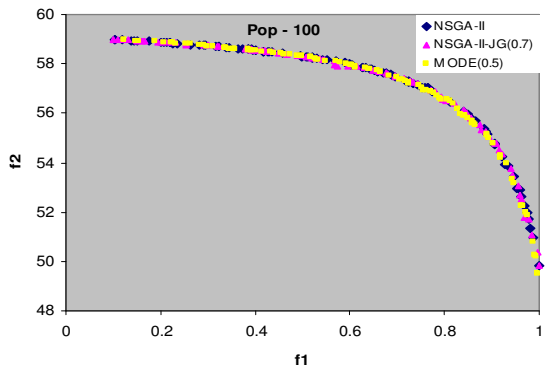


Fig. 2(c). Effects of NP and other parameters on different algorithms for CaseII

Effect of Key Parameters:

For a population size of 100, NSGA-II-JG is found to perform well (in terms of both Pareto set and diversity) with $P_{JUMP}=0.7$ and $P_{cross}=0.95$, in most of the cases. This is well evident from Table 2, Table-4 and Table-5. It is also suggested not to choose high value of P_{JUMP} , as it increases the number of generation. As far as MODE is concerned, it is observed from Table 3 and Table-5 that the following combinations of CR and F provide excellent result for the two objective problems.

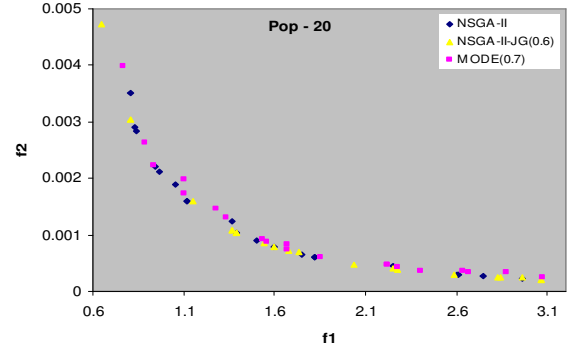


Fig. 3(b). Effects of NP and other parameters on different algorithms for Case III

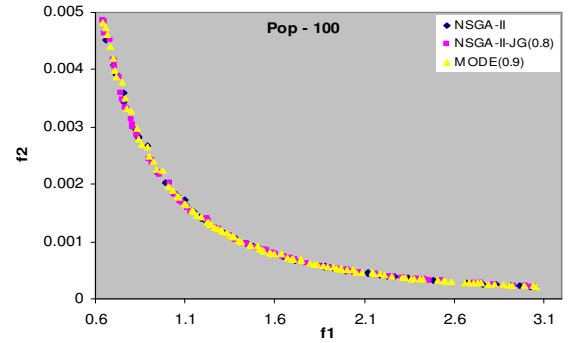


Fig. 3(c). Effects of NP and other parameters on different algorithms for Case III

Crossover Probability (CR)	Scaling Factor (F)
Low	Low
Low	High
High	Low

For any number of objective functions, it is better to choose the following range of values for F and CR: 0.5 – 0.7 and 0.7 – 0.9 respectively.

Effect of Different Strategies on MODE:

The following strategies of DE have been studied for their performance:

- I. DE/rand/1/bin
- II. DE/rand/1/exp
- III. DE/current-to-rand/1/bin

It is found that the first strategy performs better in terms of both Pareto set and diversity.

In normal DE, the trial vector replaces the parent vector in the next generation if it has better fitness value. It is not so in MODE as the child vectors are chosen from the global competition. Thus, the randomness of the DE is greatly affected and would have lead to the premature convergence of the algorithm at a low population. However, for a large population size, the algorithm could converge to global pareto set. This increases the run-time complexity of the algorithm.

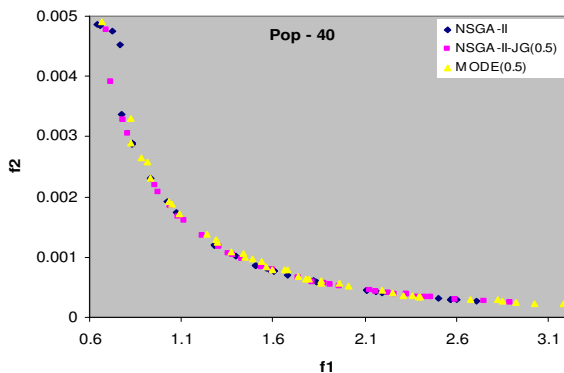


Fig. 3(a). Effects of NP and other parameters on different algorithms for Case III

8. Conclusions

The major advantage of the MODE over other existing algorithms is its diversity. Thus it is concluded that MODE could be applied to the more complicated systems where other algorithms fail to provide diverse solutions. It also provides an alternative means to solve MOPs. Yet, the run-time complexity of MODE is an area that has to be explored for further improvement.

References

- [1] K. Deb, *Multi-objective Optimization using Evolutionary Algorithms*. Chichester, U.K.: Wiley, 2001.
- [2] K. Deb, S. Agarwal, A. Pratap and T. Meyariyan, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197, April 2002.
- [3] Rahul B. Kasat and Santosh K. Gupta, "Multi-objective optimization of an industrial fluidized-bed catalytic cracking unit (FCCU) using genetic algorithm (GA) with the jumping genes operator," *Computers & Chemical Engineering*, vol. 27, no. 12, pp. 1785-1800, December 2003.
- [4] K. Price and R. Storn, "Differential Evolution – A simple evolution strategy for fast optimization," *Dr.Dobb's Journal*, vol. 22, no. 4, pp. 18-24, April 1997.
- [5] H. A. Abbass and R. Sarker, "The Pareto Differential Evolution Algorithm," *International Journal on Artificial Intelligence Tools*, Vol. 11, No. 4, pp. 531-552, December 2002.
- [6] B. V. Babu and Rakesh Angira, "New Strategies of Differential Evolution for Optimization of Extraction Process," presented at the 2003 International Symposium & 56th Annual Session of IChE (CHEMCON), Bhubaneswar.
- [7] B. V. Babu and K. K. N. Sastry, "Estimation of Heat Transfer Parameters in a Trickle Bed Reactor using Differential Evolution and Orthogonal Collocation," *Computers and Chemical Engineering*, Vol. 23 (No. 3), pp. 327-339, 1999.
- [8] B. V. Babu, *Process Plant Simulation*. New Delhi, India: Oxford University Press, 2004.
- [9] G. C. Onwubolu and B. V. Babu, *New Optimization Techniques in Engineering*. Heidelberg, Germany: Springer-Verlag, 2004.
- [10] B. V. Babu and S.A.Munawar, "Differential Evolution for the Optimal Design of Heat Exchangers", Proceedings of All India Seminar on "Chemical Engineering Progress on Resource Development: A Vision 2010 and Beyond", organized by IE (I), Orissa State Centre Bhubaneswar, March 13, 2000.
- [11] B. V. Babu and Rishindra Pal Singh, "Optimization and Synthesis of Heat Integrated Distillation Systems Using Differential Evolution", Proceedings of All India Seminar on "Chemical Engineering Progress on Resource Development: A Vision 2010 and Beyond", organized by IE (I), Orissa State Centre Bhubaneswar, March 13, 2000.
- [12] B. V. Babu and Gaurav Chaturvedi, "Evolutionary Computation strategy for optimization of an Alkylation Reaction", Proceedings of International Symposium & 53rd Annual Session of IChE (CHEMCON-2000), Science City, Calcutta, December 18-21, 2000.
- [13] B. V. Babu and Rakesh Angira, "Optimization of Non-Linear Functions Using Evolutionary Computation", Proceedings of 12th ISME Conference on Mechanical Engineering, Crescent Engineering College, Chennai, January 10-12, 2001, Paper No. CT07, pp. 153-157 (2001).
- [14] B. V. Babu and S.A.Munawar, "Optimal Design of Shell-and-Tube Heat Exchangers using Different Strategies of Differential Evolution", PreJournal.org – The Faculty Lounge, Article No. 003873, posted on website Journal <http://www.prejournal.org>, March 03, 2001.
- [15] B. V. Babu and Rakesh Angira, "Optimization of Thermal Cracker Operation using Differential Evolution", Proceedings of International Symposium & 54th Annual Session of IChE (CHEMCON-2001), CLRI, Chennai, December 19-22, 2001.
- [16] B. V. Babu and K.Gautam, "Evolutionary Computation for Scenario-Integrated Optimization of Dynamic Systems", Proceedings of International Symposium & 54th Annual Session of IChE (CHEMCON-2001), CLRI, Chennai, December 19-22, 2001.
- [17] B. V. Babu and Rakesh Angira, "A Differential Evolution Approach for Global Optimization of MINLP Problems", Proceedings of 4th Asia-Pacific Conference on Simulated Evolution And Learning (SEAL'02), Singapore, November 18 - 22, 2002, Paper No. 1033, Vol. 2, pp. 880-884, 2002
- [18] B. V. Babu and Rakesh Angira, "Optimization of Non-Linear Chemical Processes Using Evolutionary Algorithm", Proceedings of International Symposium & 55th Annual Session of IChE (CHEMCON-2002), OU, Hyderabad, December 19-22, 2002.
- [19] Rakesh Angira and B. V. Babu, "Evolutionary Computation for Global Optimization of Non-Linear Chemical Engineering Processes", Proceedings of International Symposium on Process Systems Engineering and Control (ISPSEC'03) - For Productivity Enhancement through Design and Optimization, IIT-Bombay, Mumbai, January 3-4, 2003, Paper No. FMA2, pp. 87-91 (2003).
- [20] B. V. Babu and Rakesh Angira, "Optimization of Water Pumping System Using Differential Evolution Strategies", Proceedings of The Second International Conference on Computational Intelligence, Robotics, and Autonomous Systems (CIRAS-2003), Singapore, December 15-18, 2003.
- [21] B. V. Babu, Rakesh Angira, Pallavi G.Chakole, and J.H.Syed Mubeen, "Optimal Design of Gas Transmission Network using Differential Evolution", Proceedings of The Second International Conference on Computational Intelligence, Robotics, and Autonomous Systems (CIRAS-2003), Singapore, December 15-18, 2003.
- [22] B. V. Babu, Rakesh Angira, and Anand Nilekar, "Optimal Design of an Auto-Thermal Ammonia Synthesis Reactor Using Differential Evolution", Proceedings of The Eighth World Multi-Conference on Systemics, Cybernetics and Informatics (SCI-2004), July 18-21, 2004, Orlando, Florida, USA.

Note: The soft copies of full papers in PDF format of the references in which B.V. Babu is one of the authors as listed above are available at the homepage of B.V.Babu, the URL of which is <http://discovery.bits-pilani.ac.in/discipline/chemical/BVb/publications/html>.